New XML-Based Files

Implications for Forensics

Two new office document file formats (Office Open XML and OpenDocument Format) make it easier to glean time stamps and unique document identifiers while also improving opportunities for file carving and data recovery.

Simson L. Garfinkel
US Naval Postgraduate School

James J. Migletz
US Marine Corps Command and Control Integration Division

For more than 20 years, programs such as Microsoft Word have stored their documents in binary file formats. That's changing as Microsoft, Sun Microsystems, and other developers migrate to new XML-based formats for document files.

Document files are of critical interest to forensic practitioners because of the data they contain; they're also a rich topic for forensic research. Although most investigations concern themselves solely with a document's surface content, some examinations dive deeper, examining the metadata or deleted material that's still present in the file. Investigators can, for instance, use metadata to identify individuals potentially responsible for unauthorized file modification, establish text plagiarization, or even indicate falsification of evidence. Unfortunately, metadata can also be modified to implicate innocent people—and the ease of modifying these new files means that it's far easier to make malicious modifications that are difficult (if not impossible) to detect.[1]

With so many aspects to consider, we present a forensic analysis of the two rival XML-based office document file formats: the Office Open XML (OOX) that Microsoft adopted for its Office software suite and the OpenDocument Format (ODF) used by Sun's OpenOffice software. We detail how forensic tools can exploit features in these file formats and show how these formats could cause problems for forensic practitioners. For additional information on the development and increased use of these two file formats, see the "Background" sidebar.

Analysis and Forensic Implications

To begin our analysis, we created multiple ODF and OOX files using Microsoft Office 2007 for Windows, Microsoft Office 2008 for Macintosh, OpenOffice 2.3.1, and NeoOffice 2.2.2 (a version of OpenOffice that runs under MacOS). We analyzed these files using specially written XMLdoc tools, including an XML ZIP-file browser, a search utility, and a program for automatically displaying the differences between two XML-containing ZIP files. These tools are freely available from our Web site (www.afflib.org).

For this study, we decided not to evaluate the digital signature provisions of either the ODF or OOX formats because these features are rarely used in practice.

Data Recovery

Overall, we found that ODF and OOX files tend to be smaller than equivalent legacy non-XML files, almost certainly a result of ZIP compression. Although it's trivial to add to or remove parts from a ZIP archive after its creation, we found that in many cases, adding or removing parts to the archive corrupted the file so that it couldn't be processed with Microsoft Office or OpenOffice.

The ZIP structure for these files is useful when performing data recovery or *file carving*. (File carving is the process of recognizing files by their content, rather than file system metadata. Carving is frequently used for recovering files from devices that have hardware errors, have been formatted, or have been partially overwritten.) Because each part of the archive includes a multibyte signature and a 32-bit cyclic redundancy check (CRC32) for validation, we can recover parts of

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 2009**	2. REPORT TYPE	3. DATES COVERED **00-03-2009 to 00-04-2009**
4. TITLE AND SUBTITLE **New XML-Based Files. Implications for Forensics**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Monterey,CA,93943**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

a ZIP archive even when other parts of it are damaged, missing, or otherwise corrupted. We can also use the CRC32 and relative offsets within the archive to automatically reassemble fragmented ZIP files.[2] We can then manually process recovered parts or insert them into other OOX/ODF files to view the data.

Manifest. ODF and OOX both contain a ZIP directory as the last structure in the file. We can examine this directory using standard tools, such as the Unix unzip command or Sun's JAR.

ODF has a second directory that stores document parts in an XML data structure called Meta-INF/manifest.xml. The OOX files store references to the additional document parts in the [Content_Types].xml and .rels parts, in addition to the document contents themselves.

Contents. Both file formats include a special XML file that contains the document's main flow. In ODF, the file content is called content.xml. The primary contents of an OOX word processing document created with Microsoft Office 2007 or 2008 reside in the document.xml part, although the standard allows a different name to be specified in the [Content_Types].xml part.

Forensic tools should extract text from the content parts, but tool developers must understand that text can be present in other document parts as well. For example, Microsoft Word allows other Word documents to be embedded within a Word document using the "Insert/Object..." menu command. These documents are embedded as a named .docx file inside the ZIP archive, as Figure 1 shows. In such an instance, where files are embedded within other files, investigators should analyze files recursively using a special forensic tool.

The most straightforward way for forensic practitioners to handle these new compound document formats is to save the file and then open it with a compliant program. Although this approach works, it raises several potential problems:

- The compound document might contain active content that the forensic investigator doesn't wish to execute. (Despite assurances from Microsoft and others that these file formats are safer, both ODF and OOX have provisions for storing active content[3] and therefore can carry viruses.)
- Links to external Web sites can reveal that someone has captured the file and is analyzing it.
- If parts of the file are overwritten or missing, applications such as Word or OpenOffice might be unable to open the files.
- Desktop applications can overlook or ignore critical information of interest to the forensic investigator.

```
Length   Name
-------  ----
1527     [Content_Types].xml
735       _rels/.rels
1107     word/_rels/document.xml.rels
4780     word/document.xml
6613     word/media/image1.png
7559     word/theme/theme1.xml
39832    docProps/thumbnail.jpeg
25316    word/embeddings/Microsoft_Word_Document1.docx
2036     word/settings.xml
276      word/webSettings.xml
734      docProps/app.xml
726      docProps/core.xml
15019    word/styles.xml
1521     word/fontTable.xml
-------  -------
107781   14 files
```

Figure 1. ZIP archive directory. We embedded a Microsoft Word document inside another Microsoft Word document with Word's "Insert/Object..." command.

To this end, we tested both Guidance's EnCase 6.11 and AccessData's Forensic ToolKit 1.8 and determined that they could display and search for text inside ODF files, OOX files, and OOX files embedded as objects inside other OOX files.

Both the compressed nature of ODF and OOX files and the multiple codings for the strings possible within XML represent a significant problem for forensic program developers. Because all the text is compressed, it's no longer possible to find it by scanning for strings within raw disk or document images. And because XML allows strings to be coded in hexadecimal or even interrupted by comment characters (for example, `str<--! ignore-->ing`), any forensic tool that takes shortcuts in decoding the ZIP archive or implementing the full XML schema could return false negatives when performing searches.

Embedded objects and thumbnails. A big advantage of these XML file formats is that images and other objects embedded in word processing files are stored in the ZIP file as their own parts.

We found that Microsoft Office 2008 and NeoOffice for Macintosh both stored thumbnail images of the documents' first page by default: Microsoft stores the thumbnail as a .jpg, while NeoOffice stores it as two files—a .png and a .pdf. We also found .pptx thumbnails created by PowerPoint 2007 on Windows. However, Word 2007 and Excel 2007 didn't save thumbnails by default, presumably because the

Background

Document files are fundamentally container files—that is, single files (a consecutive stream of bytes) that contain multiple data objects. A typical Microsoft Word file might contain data streams associated with the summary info, the main text, tables, and embedded images. The file also contains numerous forms of metadata—both for the document and for the container itself.

OpenDocument Format

Sun Microsystems submitted the OpenOffice OpenDocument Format (ODF) to the Organization for the Advancement of Structured Information Standards (Oasis). The ODF was approved as an Oasis standard on 1 May 2005[1] and adopted as ISO 26300 the following year.

Because of the verbose nature of XML, ODF calls for the XML file to be compressed. Parsing XML can also be time-consuming, so ODF uses a single document represented by multiple XML files bundled together into a single ZIP archive. Images and other binary objects aren't coded as XML but are stored natively as binary sections in the ZIP archive.

Microsoft's Office Open XML

Following the introduction of ODF, Microsoft introduced its own XML-based document file formats called WordprocessingML, SpreadsheetML, and PresentionML.[2] Like ODF, Office Open XML (OOX) is a ZIP archive file consisting of multiple XML document elements (unless the file is encrypted, in which case it's an OLE compound file). Microsoft refers to the file as a *package*, with each file within the archive referred to as a *part*. [3] As with ODF, structured information is first encoded into XML and compressed; embedded images are stored as binary objects within their own parts.

Because Microsoft's XML languages are defined in terms of behaviors built in to Microsoft Office, OOX files can't be readily translated into ODF or vice versa.

Microsoft's Office 2003 allowed these formats to be used as alternative document file formats; with Microsoft Office 2007, the XML-based document formats became the default file format.[3] Native support for Office Open XML is provided today in Microsoft Office 2007 for Windows and Office 2008 for Macintosh. Additionally, several other programs have the ability to read or write Word 2007 files.

Implications of ZIP for Office Documents

ZIP files consist of one or more file sections followed by a central directory. Each file section consists of a local file header that includes metadata such as the file's directory and filename, time stamp, compression method used, and additional information, followed by the actual file data and a data descriptor that includes a 32-bit checksum. The Central Directory Record contains the names of all the files, their offsets within the file, and their time stamps.[4]

The new XML-based file formats have several advantages when compared with binary file formats:

- Because they're compressed, files in the new format are typically smaller than files in the legacy format.
- Programs that process document files need only extract the sections that they're concerned with and can ignore the rest.

```
13 0 obj
<</Creator<FEFF0049006D00700072006500730073>
/Producer<FEFF004E0065006F004F006600660069006300650
0200032002E0032>
/CreationDate(D:20080311114631-07'00')>>
endobj
```

Figure 2. The header of a PDF embedded in a NeoOffice thumbnail.pdf file. The creator is the UTF-16 coding of the word "Impress," the producer is the UTF-16 coding for NeoOffice 2.2, and the creation date is 2008-03-11, 11:46:31 Pacific Daylight Time (PDT).

"Save preview picture" on the "Advanced Options" for the "Save" dialog box isn't checked by default on Word and Excel 2007 the way it is in PowerPoint.

Embedded thumbnails can be valuable in forensic practice. If the thumbnail doesn't match the document, then someone modified the thumbnail or the document after the file's creation. If the file is no longer intact, the thumbnail might give the investigator some idea of the file's contents before the file was damaged. The thumbnail can also give a sense of what the document is about if the document file itself is corrupted and can't be completely recovered.

For completeness, we also examined the thumbnail images for metadata. The .jpg thumbnails created by Microsoft Office contained metadata for only the image size and resolution, whereas the .pdf thumbnails created by NeoOffice filled in the PDF's creator, producer, and creation date. However, these values merely indicated the program that created the thumbnail, not the user who ran the program, as Figure 2 shows.

Ownership Attribution and Unique Identifiers

Unique identifiers stored within documents can play an important role in many forensic investigations. Because unique identifiers remain the same even when the document is edited, we can use them to track the movement of documents through or between organizations. By correlating unique identifi-

Table 1. Prevalence of various file types, as determined using Google.

FILE TYPE		LEGACY		OPEN OFFICE XML		OPENDOCUMENT FORMAT	
Word processing	3/08	.doc	34,900,000	.docx	29,500	.odt	54,800
	7/08	.doc	35,500,000	.docx	48,900	.odt	64,800
	9/08	.doc	41,500,000	.docx	86,100	.odt	82,500
	1/09	.doc	52,200,000	.docx	163,000	.odt	114,000
Spreadsheet	3/08	.xls	6,810,000	.xlsx	5,980	.ods	12,300
	7/08	.xls	8,510,000	.xlsx	8,880	.ods	13,400
	9/08	.xls	9,310,000	.xlsx	14,700	.ods	16,700
	1/09	.xls	11,500,000	.xlsx	52,000	.ods	24,900
Presentation	3/08	.ppt	4,790,000	.pptx	11,900	.odp	17,900
	7/08	.ppt	5,540,000	.pptx	21,000	.odp	24,800
	9/08	.ppt	6,020,000	.pptx	31,500	.odp	26,600
	1/09	.ppt	7,730,000	.pptx	57,300	.odp	37,700

- Only sections that could contain computer viruses need to be scanned for computer viruses.
- Even if parts of the file are corrupted, complete ZIP sections can still be recovered. This could allow embedded images or even content to be recovered under some circumstances.

Existing tools for handling ZIP files and XML documents make it easier for developers to write programs that can automatically process data stored in XML document files than to process legacy Word documents. However, because these are ZIP files of XML documents, they're far easier to modify. With off-the-shelf tools, an attacker can open one of these files and selectively add or remove information.

Frequency of ODF and OOX

Both ODF and OOX are still relatively rare, but their numbers are increasing. We performed Google searches by file type in March, July, and September 2008, as well as January 2009 (see Table 1), and saw the number of OOX files nearly triple during this study period.

References

1. M. Brauer et al., *Open Document Format for Office Applications v. 1.0*, Oasis specification, 1 May 2005; www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf.
2. *Overview of WordprocessingML*, tech. report, Microsoft, 2008; http://msdn2.microsoft.com/en-us/library/aa212812(office.11).aspx.
3. F. Rice, *Introducing the Office (2007) Open XML File Formats*, tech. report, Microsoft, 2008; http://msdn2.microsoft.com/en-us/library/aa338205.aspx.
4. P. Katz, *Appnote.txt—.Zip File Format Specification*, PKWare, 28 Sept. 2007; www.pkware.com/documents/casestudies/Appnote.txt.

ers found on multiple hard drives, it's possible to find previously unknown social networks.[4] We can use unique identifiers that survived copying and pasting to show plagiarism.

Unique identifiers can also raise privacy concerns. We found many unique identifiers stored within the ODF and OOX files. Some of them were "unique" in that they didn't occur elsewhere within a specific XML part or within the ZIP file: primarily, these were 32-bit numbers stored in hexadecimal. Others were 128-bit numbers unique for a particular generation of a particular document. We didn't find any unique identifiers that appeared to be unique for a specific machine.

For example, OOX defines a revision identifier for paragraphs (rsidP and rsidR; see Figure 3). Microsoft Word uses these identifiers to determine the editing session in which a user added a paragraph to the main document, to aid in Word's "Compare Documents" feature.[5] According to the specification, the rsidR values should be unique within a document: instances with the same value within a single document indicate that modifications occurred during the same editing session.

```
<w:footnote w:id="0" w:type="separator">
 <w:p w:rsidP="003A51D4" w:rsidR="003A51D4"
 w:rsidRDefault="003A51D4">
 <w:pPr>
 <w:spacing w:after="0" w:line="240"
 w:lineRule="auto"/>
```

Figure 3. Unique paragraph revision identifiers in the Microsoft OOX file format. In this case, the paragraph type appears in the footnotes.xml section.

The primary value of these identifiers to forensic examiners is document tracking. Consequently, it's possible—using these numbers—to show that one file probably resulted from editing another file (although there is, of course, a one in four billion chance that two of these 32-bit numbers will be the same). However, the new XML-based formats also make it easier to change unique IDs, making it much easier to maliciously implicate an innocent computer user or create the appearance of a false correlation.

We can imagine two uses of these unique identifiers for a forensic examiner. First, it's possible to determine the document's editing history, even if change tracking isn't enabled. Second, it should be possible to generate a database of revision identifiers that appeared in documents distributed by an organization under investigation. An automated forensic analysis tool could use such a database to generate an alert whenever an analyst discovers documents with this type of identifier on captured media or observes it traveling over a network.

To verify this hypothesis, we created a Word 2007 document with three paragraphs and saved the document. We reopened the package, added another paragraph, and saved the document again under a new name. We then repeated this process to create a third document. By examining the differences within the ZIP archives, we found the rsidR values remained consistent for the common paragraphs in each document.

Word 2007's well-hidden "Trust Center" lets users disable the storage of revision identifiers in documents by unclicking the option "Store Random Number to Improve Combine Accuracy." Microsoft's choice of language to describe this feature is quite unrevealing; a commentator named Brian Jones noted on Microsoft's blog that a better word for this option would be "Enable Anonymous Edit Tracking."

The OOX documents also include a store item ID that is used to distinguish between custom-created pieces and to ensure that data is bound to the correct location.[6] We can use this identifier in a tracking manner similar to a revision identifier.

Additionally, we found unique identifiers in each PowerPoint slideLayoutN.xml file created from both Office 2007 and 2008. People frequently create new PowerPoint presentations by starting with an old one and changing the slides. Because the PowerPoint .pptx format maintains all of the slide layouts, an investigator could trace presentations that have moved through organizations by the presence of nonstandard layouts added or by changes to the default layouts.

We weren't able to observe any unique identifiers in the ODF documents.

Time Information

Time is frequently of critical importance in forensic investigations. Even though the clock on a suspect's computer can't be trusted, they're frequently correct (especially on computers that set their time automatically over the Internet). Even when a clock is wrong, most clocks nevertheless run at more or less an even rate.[7] Thus, forensic examiners can frequently use file modification and access times to determine what files someone saw or modified within a certain time period.

Both ODF and OOX contain numerous internal time stamps indicating the time that documents were created or modified. Time stamps are present in the ZIP archive itself, in the embedded XML files, and potentially in other embedded objects (for example, in the EXIF headers of embedded JPEGs). Although these time stamps are artifacts of the ZIP file creation and aren't displayed by Office applications to the user, they nevertheless have the potential to retain some information about the ZIP file's creation and thus could be useful in a forensic investigation.

We examined the ZIP files created by NeoOffice and OpenOffice and found that the times set for the files' time stamps within the ODF ZIP archive matched the system clock. The system expressed the time in GMT, without a local time zone correction. Microsoft Word and Excel, on the other hand, set the time stamp on the files within the OOX ZIP archive to be 1 January 1980 (the epoch of the Microsoft file allocation table [FAT32] system).

In addition to these ZIP directory time stamps, we found many other time stamps embedded within various XML sections:

- Word 2007, Excel, and PowerPoint put the document's creation date in the `dcterms:created` tag of the core.xml file. All three coded the modified date in the `dcterms:modified` element of the same file.
- PowerPoint 2007 coded the document's creation date in each slideLayoutn.xml file's `a:fld` XML tag.
- When we enabled change tracking in Word 2007, the XML file annotated multiple `w:ins` tags. Each tag included a `w:author` attribute with the editor's name, a `w:date` attribute with the date of the modification, and a `w:id` attribute with the modification's ID number.
- NeoOffice encoded the document's creation date in the `meta:creation-date` tag of the meta.xml section contained within the blank OpenDocument presentation, spreadsheet, and text (ODP, ODS, and ODT) files we made.
- NeoOffice likewise embedded a thumbnail.pdf file inside the blank ODP, ODS, and ODT files we made. This .pdf file included comments for a creation date, as Figure 2 shows.
- NeoOffice embedded the date in a `text:date` tag within the styles.xml of a blank presentation we made.

These time stamps might be significant in a forensic examination—for example, they might show when someone edited an ODF or OOX file with an ODF/OOX-aware application. Or the time stamps might indicate multiple editing sessions. Alternatively, they might indicate a tampered document.

The time stamps of ODF files are particularly relevant to file carving because all the time stamps for files within each ZIP archive are the same and, in practice, will likely differ from the time stamps in other ZIP archives on the same hard drive. Consequently, we can use these time stamps as a unique identifier for a particular ODF file, which in turn allows us to find fragmented ODF files even when the files are corrupt; in many cases, we can intelligently reassemble them.

Hiding Data

There are several approaches for hiding data in ODF and OOX files from forensic analysis, including traditional encryption and the use of comment fields.

Encryption

Both OpenOffice and Microsoft Office make it possible to save files with encryption so that the user must provide a password to open a document. We created encrypted documents containing simple text and with embedded photographs using both OpenOffice and Microsoft Office.

ODF and OOX take fundamentally different approaches to encrypting documents. ODF applies encryption to some of the specific segments of the document file, leaving other segments unencrypted. The ZIP metadata and directory information, for example, remain unencrypted. However, all the document content parts are encrypted, including these ZIP sections:

- configurations2/accelerator/current.xml,
- content.xml,
- settings.xml,
- styles.xml,
- pictures/image.png,
- thumbnails/thumbnail.pdf, and
- thumbnails/thumbnail.png.

These sections were present and unencrypted:

- Meta-INF/manifest.xml,
- meta.xml, and
- mimetype.

The lack of encryption could potentially leak the values for XML tags that might be relevant to an investigation:

- `meta:generator`—the specific build of the specific application that created the document;
- `meta:creation-date`—the document's creation date in local time;
- `dc:language`—the document's primary language;
- `meta:editing-cycles`—the number of times someone edited the document;
- `meta:user-defined`—user-definable metadata (in our case, tags with the values Info 1 through Info 4); and
- `meta:document-statistics`—including the number of tables, images, objects, page count, paragraph count, word count, and character count.

The OOX format stores encrypted files as OLE compound files that have the same .docx, .xlsx, or .pptx file extension.[8] The popular 7-Zip decompression application (www.7-zip.org) can read these files and extract the following encrypted segments:

```
[6]DataSpaces
[6]DataSpaces/DataSpaceInfo
[6]DataSpaces/DataSpaceInfo/
   StrongEncryptionDataSpace
[6]DataSpaces/DataSpaceMap
[6]DataSpaces/TransformInfo
[6]DataSpaces/TransformInfo/
   StrongEncryptionTransform
[6]DataSpaces/TransformInfo/
   StrongEncryptionTransform/[6]Primary
[6]DataSpaces/Version
EncryptedPackage
EncryptionInfo
WordDocument
```

Examining these files revealed their encryption using the Rivest, Shamir, and Adleman (RSA) and Advanced Encryption Standard (AES) algorithms with a 128-bit key. We didn't attempt further cryptanalysis.

Hiding Data in Comments

We identified several opportunities for hiding data in ODF and OOX files using comments. The first approach doesn't work well; the second and third do:

1. *Adding sections to the ZIP archive.* When Microsoft Word encounters a file modified in this manner, it reports that the archive is corrupted and provides an option to recover its data. NeoOffice, on the other hand, silently ignores these additional sections and opens the ODF file.
2. *Placing comments directly into the ZIP archive using the comment feature that the ZIP file format provides.* We tested Office 2007, 2008, and NeoOffice and found that these programs appear to ignore ZIP comments when the files are read. When the files are written back out, the application strips the comments.
3. *Adding comments to the XML files as XML comments.* The programs we tested ignored the data stored as XML comments. Once again, when files with

```
<a:fld id=''{985AE863-DF53-4B19-9956-91DEFC2F01C1}''
type=''datetimeFigureOut''>
```

Figure 4. A "unique" identifier. This tag in a Microsoft PowerPoint slide's XML section appears unique, but is it really?

comments are written back out, the application strips them.

These comment fields are of concern for forensic analysis because, while many document formats allow for embedded comments, the forensic tools we tested ignore comments stored in ODF and OOX files. Someone can therefore use embedded comments as a channel for covert communications, which is particularly attractive because processing the file with Word may inadvertently remove the covert information.

We created a tool called docx-steg.py for hiding arbitrary files in Microsoft .docx documents and recovering them at a later time. Our tool takes the file, encrypts it, base64 encodes it, and stores it as a comment inside one of the file's XML sections. This tool is also available for download from our Web site.

Despite the fact that Sun Microsystems and Microsoft have submitted the ODF and OOX specifications to standards bodies, surprisingly few technical articles have published details about the new XML document file formats, and virtually nothing has been published regarding their forensic implications. This might be in a case where the forensic analysis depends on particular aspects of the XML or specific application software behaviors.

XML is frequently called self-documenting, but that's only the case when tags and attributes have names that are easy to figure out. Unfortunately, most information gleaned from inspection is unlikely to withstand scrutiny in court: opposing counsel will almost certainly demand a clear formal documentation if a case depends on an XML tag's interpretation. Consider the XML fragment we show in Figure 4. Just because the `a:fld` tag appears to have a unique identifier in its `id=` attribute, it isn't enough to establish that it is, in fact, a unique identifier.

Despite this important caveat, new XML-based file formats offer many opportunities for forensic examiners to learn about the process by which someone created a specific document. Although the information in these new XML files might also reside in legacy .doc, .xls, and .ppt file formats, the move to XML makes this information far more readily accessible. It's our hope that a new generation of forensic tools can make programmatic use of this information without the need for additional human intervention. □

Acknowledgments

We thank Jessy Cowan-Sharp, George Dinolt, Beth Rosenberg, and the anonymous reviewers for their comments on previous versions of this article. This work was funded in part by the US Naval Postgraduate School's Research Initiation Program. The views and opinions expressed in this document represent those of the authors and do not necessarily reflect those of the US government or the Department of Defense.

References

1. S. Rodriguez, "Microsoft Office XML Formats? Defective by Design," Aug. 2007; http://ooxmlisdefectivebydesign.blogspot.com/2007/08/microsoft-office-xml-formats-defective.html.
2. S. Garfinkel, "Carving Contiguous and Fragmented Files with Fast Object Validation," *Digital Investigation*, 2007; www.dfrws.org/2007/proceedings/p2-garfinkel.pdf.
3. P. Lagadec, "Openoffice/Opendocument and MS Office 2007/Open XML Security," *Proc. Pacific Security Conf.*, 2006; http://pacsec.jp/psj06/psj06lagadec-e.pdf.
4. S. Garfinkel, "Forensic Feature Extraction and Cross-Drive Analysis," *Proc. 6th Ann. Digital Forensic Research Workshop,* Elsevier, 2006; www.dfrws.org/2006/proceedings/10-Garfinkel.pdf.
5. *Ecma-376 Standard, Office Open XML File Formats*, Dec. 2008; www.ecma-international.org/publications/standards/Ecma-376.htm.
6. P. Aven, "A Final 'Word': Part 6 in a Series on Marklogic Servers and Office 2007," 22 Jan. 2008; http://xqzone.marklogic.com/columns/smallchanges.
7. F. Buchholz and B. Tjaden, "A Brief Study of Time," *Proc. 7th Ann. Digital Forensics Research Workshop*, Elsevier, 2007; www.dfrws.org/2007/proceedingsp31-buchholz.pdf.
8. *[Ms-offcrypto]: Office Document Cryptography Structure Specification*, Microsoft, 27 June 2008; http://msdn.microsoft.com/en-us/library/cc313071.aspx.

***Simson L. Garfinkel** is an associate professor at the US Naval Postgraduate School in Monterey, California, and an associate of the School of Engineering and Applied Sciences at Harvard University. His research interests include computer forensics, the emerging field of usability and security, personal information management, privacy, information policy, and terrorism.*

***James J. Migletz** is an enterprise architect for the Operational Architecture Branch of the US Marine Corps Command and Control Integration Division. His research interests include network security and computer forensics. Migletz has an MS in computer science from the Naval Postgraduate School. He graduated with honors and received the Admiral Grace Murray Hopper award for academic achievement.*